

- 6 -

processing an incoming call based on an audio input from a given user;

identifying a request for audio information associated with a text-based document;

searching a cache for executable code associated with the requested audio information, the executable code generated in response to a previous request from another user for audio information associated with the text-based document; and

executing corresponding executable code from the cache to satisfy the request for audio information associated with the incoming call.

REMARKS

Claims 1-19 were previously pending in the present application. Claims 20-30 have been added by way of this Amendment. Claims 6, 8-10, and 15-19 have been canceled. Accordingly, claims 1-5, 7, 11-14 and 20-30 are now pending.

Applicants have amended claims 1-5, 7, and 11-14 to expedite prosecution of the present application. No new matter was added when amending the claims. The submission of these amendments should not be interpreted as acquiescing to the Examiner's rejection.

The following remarks address the rejections of claims 1-19 as set out by Examiner in this Office Action and patentability of newly added claims 20-30.

Objection of Claims 8 and 17

The Examiner has objected to claims 8 and 17 because they were duplicates of each other. Applicants have canceled claims 8 and 17 as well as other duplicate claims 9-10 and 18-19.

Rejections of Claims 1-19 under 35 U.S.C. § 102

The Examiner has rejected claims 1-19 under 35 U.S.C. § 102(e) based on the teachings of Saylor, et al., (U.S. Patent 6,501,832).

Applicants have amended claim 1 to expedite prosecution of the present application. For example, claim 1 now recites that the compiler operates to compile a document (as retrieved by a fetcher) into compiled document data in executable form. The compiled document data is stored in a cache prior to a request for the text-based document. Support for amendments to the claim can be found in FIGS. 2 and 3, text at page 4 paragraph 26-27, page 5 paragraphs 31-34, and elsewhere throughout the specification. No new matter has been added as a result of amending claim 1.

It is well accepted that a claim is not anticipated under 35 U.S.C. § 102 unless each and every aspect of the claimed invention is taught by a single reference. Additionally, a claim is not obvious under 35 U.S.C. § 103(a) unless the cited references teach or suggest every claim limitation and combination thereof. Accordingly, it is respectfully submitted that the invention as recited in amended claim 20 is neither anticipated nor obvious because it includes distinguishing limitations not taught or suggested by any of the cited references.

For example, Saylor '832 discloses a technique of enabling users to access information using a voice code (Vcode), which is a code assigned to a particular page of content that is delivered to a user via a phone (column 1, lines 58-61). Based on a Vcode associated with requested content, a Vpage (web page) is retrieved from Vpage (web page) database 18 or retrieved over a network 20 from Vpage servers 22 (column 18, lines 23-27). In general, a Vpage is a web page such as that based on VoiceXML.

Vpages (web pages in Voice XML format) are stored in database 18 where they are retrieved as shown in FIG. 2. During operation, Vpage retrieval system 32 retrieves unprocessed Vpages (web pages) that are in turn processed in response to the Vcode by Vpage menu module 36 and Vpage execution module 34.

As discussed in more detail in corresponding text at column 18, lines 45-65, upon retrieval of the Vpage corresponding to the Vcode, Vpage execution module 34 thereafter interprets retrieved Vpage (webpage) content provided by Vpage (web page) retrieval system 32. As discussed in Saylor '832, execution (i.e., XML interpretation) of the web page content includes scanning the Vpage for certain (XML) tags (of content) and generating menus depending on the Vpage. This conventional technique is basically described in the background of the present application except, in Saylor '832, the web pages or Vpages in unprocessed XML format are optionally stored and retrieved from a local data base 18 instead of a remote server.

In a nutshell, voice network access provider (VNAP) 12 and, more specifically, Vpage (web page) menu module 36 in FIG. 2 utilizes the collection of Vpages (or web pages in XML format) stored in database 18 to generate an appropriate audible response in real-time to user 14. Note that Vpage menu module 36 includes a VoiceXML interpretation module to produce the appropriate audible response (column 18, lines 59-65). Accordingly, in Saylor, all Vpages are processed (i.e., executed) in real-time by the Vpage module 36 and Vpage execution module 34 in response to a Vcode received from a user during a phone call.

In contradistinction to this cited technique in Saylor '832, claim 1 recites that a compiler converts documents (such as web page documents originally in

XML format) into compiled document data (such as executable code) that is stored in a cache. The fetcher retrieves the appropriate compiled document data from the cache to satisfy a user request. More specifically, an execution thread services the user request by executing the compiled document data stored in cache. Based on this technique of storing pre-processed data (such as XML interpreted pages), a proper response such as an audio reply can be more quickly generated for a particular user because the executable file associated with a requested document (such as an VoiceXML page) need only be retrieved from cache and executed by execution thread to generate the response. Thus, execution thread can skip a step of processing or interpreting raw document data such as a Voice XML web page before responding to an incoming request. According to Saylor '832, a Vpage must be 'interpreted' or 'compiled' to generate an executable file (such as an audio file) each time a user requests a particular document (e.g., during a phone call).

The Examiner has likened the 'compiler' in claim 1 to content interpreter 66 (FIG. 3) as discussed in Saylor at column 21 lines 20-22. It is respectfully submitted that although Vpage server 22 does include interpreter 66, there is no mention that 'interpreted' or 'compiled' documents (or Vpages) generated by interpreter 66 are stored or cached in memory such as database 18. Instead, the raw uninterpreted Vpages (XML web page documents) themselves are stored in database 18. As its name suggests and according to the associated text in Saylor, Vpage (webpage) retrieval system 32 (FIG. 2) only retrieves Vpages (webpages) from database 18 or web page server 22 that thereafter must be processed. Saylor does not teach, disclose or suggest storing already interpreted document data as executable code as in the claimed invention. For example, already interpreted or compiled documents in executable form is stored in a cache so that an execution thread need only execute the code in response receiving to a call from a user.

- 10 -

It is respectfully submitted that in view of the above amendment and remarks, claim 1 is novel and non-obvious as it incorporates techniques contrary to previously accepted wisdom and blueprints for the inventive method cannot be found in the individual or combined cited references. Accordingly, it is submitted that independent claim 1 and corresponding dependent claims 2-4 are in condition for allowance over the prior art.

Similar amendments have been made to independent method claim 11. For applicable reasons, it is submitted that independent claim 11 and corresponding dependent claims 12-14 are in condition for allowance.

Claim 3 has been amended to recite that the voice response system of the present invention includes a backup interpreter that otherwise provides a response to a user in the event of a failure. The Examiner has likened the 'backup interpreter' in claim 3 to Vpage (webpage) menu module 36 (FIG. 2) as discussed in Saylor at column 18 lines 56-65. It is respectfully submitted that this cited passage does not discuss the claimed technique of (nor does it appreciate the technical hurdles associated with) providing a backup or secondary interpreter to provide a response in the event of a failure. In fact, Saylor does not even address the issue of how to provide continued service in the event of a failure. The technique of providing backup in the claimed invention increases overall reliability of the voice response system in the event of failures. Allowance of claim 3 is also respectfully requested.

The Examiner has also rejected claim 5 based on the teachings of Saylor '832. Claim 5 has been amended to recite that the interactive voice response system of the present invention includes a backup interpreter that otherwise provides a response to a user in the event of a failure. As mentioned, the Examiner has likened the 'backup interpreter' to Vpage (webpage) menu module 36 (FIG. 2) as discussed in Saylor at column 18 lines 56-65. It is respectfully

requested submitted that this cited passage does not discuss the claimed technique of (nor does it appreciate the technical hurdles associated with) providing a backup or secondary interpreter to provide a response in the event of a failure. In fact, Saylor does not even address the issue of how to provide continued service in the event of a failure. The technique of providing backup in claim 5 increases overall reliability of the voice response system in the event of failures. Allowance of claim 5 and corresponding dependent claim 7 is also respectfully requested. Method claim 14 also has been amended. Allowance is respectfully requested.

Patentability of New Claims 20-30

Newly submitted claim 20 includes a limitation that a cache stores executable code associated with text-based documents to satisfy a user request. Support for the claim can be found in FIGS. 2 and 3, text at page 4 paragraph 26-27, page 5 paragraphs 31-34, and elsewhere throughout the specification. No new matter is being added as a result of adding this claim.

As discussed above, voice network access provider 12 in FIG. 2 of Saylor '832 does not store executable code associated with a webpage document in memory. Thus, claim 20 is distinguished and advantageous over the cited references for applicable reasons as discussed above. Allowance of claim 20 and corresponding dependent claims 21-29 is respectfully requested.

Newly submitted claim 21 further distinguishes the claimed invention (of claim 20) over the prior art because it recites that a compiler converts the text-based document into executable speech code for storage in the cache prior to receipt of the incoming request. Support for claim 22 can be found in FIG. 2 and 3, related text at page 4 paragraph 26-27, page 5 paragraphs 31-34, and elsewhere throughout the specification. No new matter is being added as a result of adding this claim. Based on this technique, compiled or executable

code associated with a previously requested text-based document is stored in cache for later retrieval. Thus, a text-based document such as a web page need not be completely recompiled when another user requests the same text-based document. Rather, the previously compiled executable code is retrieved from cache and executed to service the new request.

Newly submitted claim 22 further distinguishes the claimed invention (of claim 20) over the prior art because it recites that the fetcher initiates a communication with a remote server in order to retrieve a text-based document such as a web page if a corresponding executable code is not stored in the cache. Support for claim 22 can be found in FIG. 2 and 3, related text at page 4 paragraph 26-27, page 5 paragraphs 31-34, and elsewhere throughout the specification. No new matter is being added as a result of adding this claim.

As discussed, Saylor does not search a cache for executable code to service a user. Nor does Saylor initiate communication with a remote server if executable code is not stored in cache. In contradistinction to Saylor, the fetcher in claim 22 initiates communication with a remote server to retrieve a document such as raw web page content. This conditional procedure of retrieving text-based documents from a remote server increases flexibility because the claimed voice response system services a request by retrieving an executable file from cache or retrieving raw document data from a remote server. Thus, the technique as recited in claim 22 is distinct and advantageous over the cited prior art.

Newly submitted claim 23 further distinguishes the claimed invention (as in claim 22) over the prior art because it recites that a compiler converts retrieved text-based documents into corresponding executable code that is stored in cache. Based on this technique of pre-compiling a retrieved text-based document, fetcher need only retrieve and execute the executable code to service

another user requesting a same text-based document as already processed by the compiler. Saylor does not address this technical hurdle anywhere in his issued patent. Thus, it is respectfully submitted that claim 23 is neither anticipated nor obvious in light of the prior art. Consideration and allowance of new dependent claim 23 is also respectfully requested.

Newly submitted claim 24 further distinguishes the claimed invention (as in claim 20) over the prior art because it recites that executable code (or compiled documents) in cache is utilized by multiple execution threads to provide a response to multiple users. Based on this technique of compiling a retrieved text-based document, fetcher need only retrieve and execute the executable code in cache in the event that multiple users simultaneously request a particular text-based document. Saylor also does not address this technical hurdle of simultaneously servicing multiple users using a common executable code in cache. Thus, it is respectfully submitted that claim 23 is neither anticipated nor obvious in light of the prior art. Consideration and allowance of new dependent claim 23 is also respectfully requested.

Conclusion

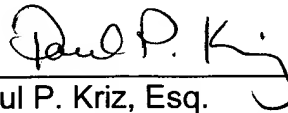
In view of the foregoing remarks, this Application should be in condition for allowance. A Notice to this affect is respectfully requested. If the Examiner believes, after this Amendment, that the Application is not in condition for allowance, the Examiner is respectfully requested to call the Applicants' Representative at the number below.

Applicants hereby petition for any extension of time which is required to maintain the pendency of this case. If there is a fee occasioned by this response, including an extension fee, that is not covered by an enclosed check, please charge any deficiency to Deposit Account No. 50-0901.

- 14 -

If the enclosed papers or fees are considered incomplete, the Patent Office is respectfully requested to contact the undersigned Attorney at (508) 366-9600, in Westborough, Massachusetts

Respectfully submitted,



Paul P. Kriz, Esq.
Attorney for Applicants
Registration No.: 45,752
CHAPIN & HUANG, L.L.C.
Westborough Office Park
1700 West Park Drive
Westborough, Massachusetts 01581
Telephone: (508) 366-9600
Facsimile: (508) 616-9805

Attorney Docket No.: NMS03-05

Dated: May 27, 2003

APPENDIX
MARKED UP VERSION OF AMENDMENTS

Amendments to the Claims Under 37 C.F.R. § 1.121(c)(1)(ii)

Please amend the claims as follows:

1. (Amended) [A VoiceXML interpreting] An interactive voice response system [including a VoiceXML Interpreter] comprising:
[a Fetcher operative to retrieve documents;]
a compiler operative to compile documents retrieved by a fetcher
into compiled document data in executable form [said Fetcher]; [and]
a cache which stores the compiled document data prior to receipt of
audio input from a given user requesting a text-based document
[documents compiled by said compiler]; and
an execution thread that executes compiled document data
retrieved from the cache by the fetcher.
2. (Amended) A [VoiceXML interpreting] system according to claim 1 [and]
also comprising a storage device [Storage Device], which stores state
information[,] related to execution of said compiled document data
[documents].
3. (Amended) A [VoiceXML interpreting] system according to claim 2 [and]
also comprising a backup VoiceXML Interpreter communicating with said
storage device, the backup interpreter providing a response to a user in
the event of a failure associated with a primary voice response system
[Storage Device].

4. (Amended) A [VoiceXML interpreting] system according to claim 3, [and] wherein said storage device [Storage Device] comprises a memory database external to said backup VoiceXML Interpreter [and to said backup Interpreter].
5. (Amended) [A VoiceXML interpreting] An interactive voice response system comprising:
 - a fetcher [Fetcher] operative to retrieve a compiled document [documents]; [and]
 - a storage device [Storage Device] which stores state information related to execution of said compiled document [documents]; and
 - a backup Voice XML interpreter in communication with said storage device utilizing the state information to execute the compiled document in the event of a failure.
7. (Amended) [A VoiceXML interpreting system] A system as in claim 5, [according to claim 6 and] wherein said storage device is [Storage Device comprises] a memory database external [to said VoiceXML Interpreter and] to said backup VoiceXML interpreter [Interpreter].
11. (Amended) [A VoiceXML interpreting] In an interactive voice response system, a method [including VoiceXML interpreting] comprising:
 - retrieving documents encoded according to VoiceXML;
 - compiling the retrieved documents into compiled document data in executable form; [and]
 - caching the compiled document data for later retrieval and execution [documents by said compiler].
12. (Amended) A [VoiceXML interpreting] method [according to] as in claim 11 [and also] further comprising:

storing state information related to execution of said compiled
document data [documents].

13. (Amended) A [VoiceXML interpreting] method [according to] as in claim 12
[and also] further comprising:

[employing] providing a backup VoiceXML Interpreter [for receiving]
that utilizes the stored state information to support continued service in the
event of a failure.

14. (Amended) [A VoiceXML interpreting] In an interactive voice response
system, a method comprising:

retrieving executable code derived from a text-based document
[documents]; and

storing state information related to execution of the compiled
document; and

providing the state information to a backup VoiceXML interpreter to
otherwise provide a response to a user in the event of a failure [said
documents].